

## Lab 1

**Reg. No:** 19BCE1209

**Name:** Gautam Sanjay Wadhwani

**Course:** CSE4001 Parallel and Distributed Computing

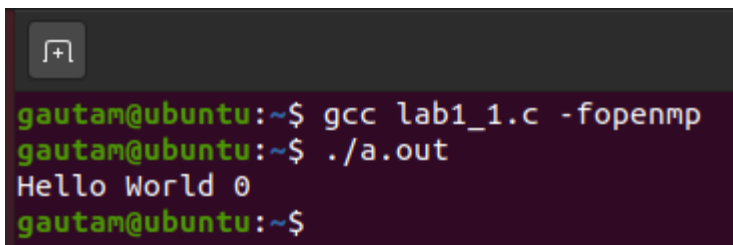
**Q1.** Parallel program to print hello world

**Code:**

```
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <omp.h>

int main() {
    omp_set_num_threads(1);
    #pragma omp parallel
    {
        int id = omp_get_thread_num();
        printf("Hello World %d\n", id);
    }
}
```

**Output:**

A terminal window with a dark background and light green text. It shows the compilation of 'lab1\_1.c' using 'gcc' with the '-fopenmp' flag, followed by running the resulting executable './a.out'. The output is 'Hello World 0'.

```
gautam@ubuntu:~$ gcc lab1_1.c -fopenmp
gautam@ubuntu:~$ ./a.out
Hello World 0
gautam@ubuntu:~$
```

**Q2.** Parallel program to print hello world (2 threads and 4 threads)

**Code:**

```
#include <stdio.h>

#include <pthread.h>

#include <stdlib.h>

#include <omp.h>

int main() {

    omp_set_num_threads(4);

    #pragma omp parallel

    {

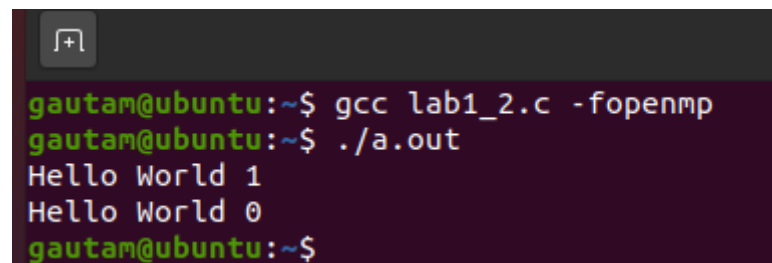
        int id = omp_get_thread_num();

        printf("Hello World %d\n", id);

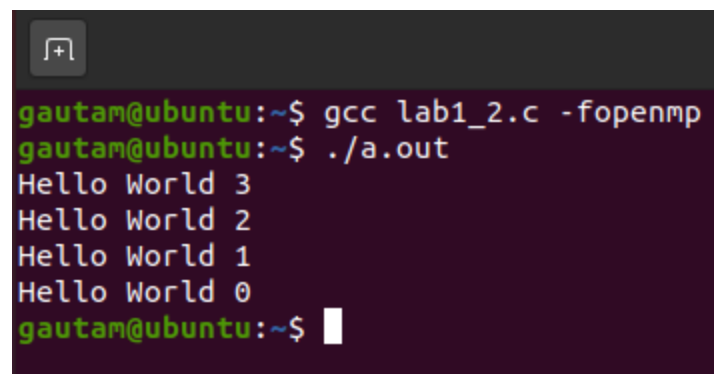
    }

}
```

**Output:**



```
gautam@ubuntu:~$ gcc lab1_2.c -fopenmp
gautam@ubuntu:~$ ./a.out
Hello World 1
Hello World 0
gautam@ubuntu:~$
```



```
gautam@ubuntu:~$ gcc lab1_2.c -fopenmp
gautam@ubuntu:~$ ./a.out
Hello World 3
Hello World 2
Hello World 1
Hello World 0
gautam@ubuntu:~$
```

**Q3. Parallel program to add two arrays a and b**

**Code:**

```
#include <stdio.h>

#include <pthread.h>

#include <stdlib.h>

#include <omp.h>

#include <sched.h>


int main() {

int a[10], b[10], c[10];

int i;

printf("Enter values of a[i] and b[i]\n");

for(i = 0; i < 10; i++) {

scanf("%d %d", &a[i], &b[i]);

}

#pragma omp parallel

{

for (i = 0; i < 10; i++)

{

c[i] = a[i] + b[i];

printf("CPU %d\tThread %d\tValue %d\n", sched_getcpu(), omp_get_thread_num(), c[i]);

}

}

printf("Values of c[i]\n");

for(i = 0; i < 10; i++) {

printf("%d\n", c[i]);

}

return 0;

}
```

Output:

```
gautam@ubuntu:~$ gcc lab1_3.c -fopenmp
lab1_3.c: In function 'main':
lab1_3.c:19:41: warning: implicit declaration of function
   19 | printf("CPU %d\tThread %d\tValue %d\n", sched_get
      |                                           ^~~~~~
      |                                           sched_get
gautam@ubuntu:~$ export OMP_NUM_THREADS=4
gautam@ubuntu:~$ ./a.out
Enter values of a[i] and b[i]
1 2
3 4
5 6
7 8
1 4
1 9
2 9
2 5
3 7
3 8
CPU 0   Thread 0       Value 3
CPU 3   Thread 3       Value 3
CPU 3   Thread 3       Value 7
CPU 3   Thread 3       Value 11
CPU 3   Thread 3       Value 15
CPU 3   Thread 3       Value 5
CPU 3   Thread 3       Value 11
CPU 3   Thread 3       Value 7
CPU 3   Thread 3       Value 10
CPU 3   Thread 3       Value 11
CPU 2   Thread 2       Value 3
CPU 0   Thread 0       Value 10
CPU 1   Thread 1       Value 3
Values of c[i]
3
7
11
15
5
10
11
7
10
11
gautam@ubuntu:~$
```